Claims

1. (**Currently Amended**) A computerized method of creating test coverage for non-deterministic programs within a testing environment comprising:

in a computer, receiving a graph of edges and states representing a program under test, the states comprising ~~**at least one**~~ **a plurality of** deterministic state**s** controllable by the testing environment and a plurality **of** non-deterministic states uncontrollable by the testing environment**, wherein a plurality of the edges represent non-deterministic choices associated with at least one of the non-deterministic states and the edges representing non-deterministic choices are assigned respective probabilities**;

creating a continuous cycle of edges and states through the graph that reaches each state in the graph at least once;

splitting the continuous cycle into discrete sequences that end at the non-deterministic states**;**

executing the program under test **under test conditions** as a first execution of the program;

determining discrete sequences not reached by the first execution of the program;

determining untested states as states in the discrete sequences not reached by the first execution of the program;

~~**calculating, for at least some deterministic states, a probability that during program execution, a path from a given deterministic state will reach a given untested state;**~~

~~**calculating, for the at least some deterministic states, a number of edges between the given deterministic state and the corresponding untested state as a cost;**~~

creating strategies through the graph~~**, for at least of the given deterministic states,**~~ to reach ~~**the corresponding**~~ **the** untested state**s, wherein creating strategies comprises (a), (b), and (c):**

(a) starting from a given untested state out of the untested states, traversing the graph backwards and computing costs and probabilities at states with a

respective edge reaching the given untested state, wherein probabilities are assigned based on likelihood that an edge exiting a respective non-deterministic state uncontrollable by the testing environment will be selected during execution;

(b) assigning probabilities and costs to respective vertices based on a probability of providing a path to the given untested state; and

(c) choosing vertices such that a next state with a lower cost and higher probability is preferred over a next state with higher cost and lower probability**, wherein the strategies comprise respective series of one or more edge transitions through the graph**;

storing a representation of the created strategies in computer memory; and

in the computer, executing the program under test under test conditions **as a subsequent execution of the program to the first, the subsequent execution** using the ~~stored~~ created strategies such that the **subsequent** ~~program~~ execution has a higher probability than the first execution of the program to execute through states that correspond to the untested states**, wherein the subsequent execution repeats at least one of the strategies and presents the non-deterministic choices from at least one of the non-deterministic states a plurality of times during the subsequent execution, whereby a possibility of reaching at least one of the untested states is increased**.

2. (Canceled)

3. **(Currently Amended)** The method of claim 1 wherein the continuous cycle of edges is created from the graph ~~input~~ using a Chinese Postman tour algorithm.

4. **(Currently Amended)** The method of claim 1 wherein the **states of the** graph ~~states~~ are received as a set of deterministic vertices and a set of non-deterministic vertices.

5. (Canceled)

6. (Canceled)

7. (Canceled)

8. **(Currently Amended)** The system of claim ~~7~~ 27 wherein ~~a~~ the continuous cycle is determined according to a Chinese Postman algorithm.

9. **(Currently Amended)** The system of claim ~~7~~ 27 wherein the discrete sequences comprise beginning states reachable from edges exiting non-deterministic states.

10. **(Currently Amended)** The system of claim ~~7~~ 27 wherein ~~**an untouched**~~ **a** discrete sequence **not reached by the first execution of the program** is a state selectable from a program code executing at a remote computer.

11. (Canceled)

12. (Canceled)

13. (Canceled)

14. (Canceled)

15. (Canceled)

16. (Canceled)

17. (Canceled)

18. (**Currently Amended**) The computer-readable **storage** medium of claim ~~15~~ **28** wherein **at least one of** the non-deterministic **states represents** behavior ~~**comprises**~~ **comprising** communications with a remote computer.

19. (Canceled)

20. (Canceled)

21. (Canceled)

22. (**Currently Amended**) The method of claim 1 wherein ~~**the calculating**~~ **assigning** the probability comprises determining ~~**the**~~ **a** number of edges leaving the non-deterministic state as k, and calculating the probability as 1/k.

23. (Canceled)

24. (Canceled)

25. (Canceled)

26. (Canceled)

27. (New) A computer system comprising:

memory and a central processing unit executing computer-executable instructions for creating test coverage for non-deterministic programs within a testing environment, the computer-executable instructions operating to:

receive a graph of edges and states representing a program under test, the states comprising a plurality of deterministic states controllable by the testing environment and a plurality of non-deterministic states uncontrollable by the testing environment, wherein a plurality of the edges represent non-deterministic choices associated with at least one of the non-deterministic states and the edges representing non-deterministic choices are assigned respective probabilities;

create a continuous cycle of edges and states through the graph that reaches each state in the graph at least once;

split the continuous cycle into discrete sequences that end at the non-deterministic states;

execute the program under test under test conditions as a first execution of the program;

determine discrete sequences not reached by the first execution of the program;

determine untested states as states in the discrete sequences not reached by the first execution of the program;

create strategies through the graph to reach the untested states, wherein creating strategies comprises (a), (b), and (c):

(a) starting from a given untested state out of the untested states, traversing the graph backwards and computing costs and probabilities at states with a respective edge reaching the given untested state, wherein probabilities are assigned based on likelihood that an edge exiting a respective non-deterministic state uncontrollable by the testing environment will be selected during execution;

(b) assigning probabilities and costs to respective vertices based on a probability of providing a path to the given untested state; and

(c) choosing vertices such that a next state with a lower cost and higher probability is preferred over a next state with higher cost and lower probability, wherein the strategies comprise respective series of one or more edge transitions through the graph;

store a representation of the created strategies in the memory; and

execute the program under test under test conditions as a subsequent execution of the program to the first, the subsequent execution using the created strategies such that the subsequent execution has a higher probability than the first execution of the program to execute through states that correspond to the untested states, wherein the subsequent execution repeats at least one of the strategies and presents the non-deterministic choices from at least one of the non-deterministic states a plurality of times during the subsequent execution, whereby a possibility of reaching at least one of the untested states is increased.

28. (New) A tangible computer-readable storage medium having thereon computer-executable instructions for performing a method of creating test coverage for non-deterministic programs within a testing environment, the method comprising:

in a computer, receiving a graph of edges and states representing a program under test, the states comprising a plurality of deterministic states controllable by the testing environment and a plurality of non-deterministic states uncontrollable by the testing environment, wherein a plurality of the edges represent non-deterministic choices associated with at least one of the non-deterministic states and the edges representing non-deterministic choices are assigned respective probabilities;

with a Chinese Postman tour technique, creating a continuous cycle of edges and states through the graph that reaches states in the graph at least once;

splitting the continuous cycle into discrete sequences so the discrete sequences end with a non-deterministic state;

executing the program under test under test conditions as a first execution of the program;

determining discrete sequences not reached by the first execution of the program;

determining untested states as states in the discrete sequences not reached by the first execution of the program;

creating strategies through the graph to reach the untested states, wherein creating strategies comprises (a), (b), and (c):

(a) starting from a given untested state out of the untested states, traversing the graph backwards and computing costs and probabilities at states with a respective edge reaching the given untested state, wherein probabilities are assigned based on likelihood that an edge exiting a respective non-deterministic state uncontrollable by the testing environment will be selected during execution;

(b) assigning probabilities and costs to respective vertices based on a probability of providing a path to the given untested state; and

(c) choosing vertices such that a next state with a lower cost and higher probability is preferred over a next state with higher cost and lower probability, wherein the strategies comprise respective series of one or more edge transitions through the graph;

storing a representation of the created strategies in computer memory; and

in the computer, executing the program under test under test conditions as a subsequent execution of the program to the first, the subsequent execution using the created strategies such that the subsequent execution has a higher probability than the first execution of the program to execute through states that correspond to the untested states, wherein the subsequent execution repeats at least one of the strategies and presents the non-deterministic choices from at least one of the non-deterministic states a plurality of times during the subsequent execution, whereby a possibility of reaching at least one of the untested states is increased.